3. Lab Series – Threads and Processes in Windows 2000

To be handed in at Exercise/Lecture on Dec. 5, 2001.

Assignment 3.1:

Explain the approach to thread scheduling in the Windows 2000 operating system.
What is the unit of scheduling under Windows 2000?
Which scheduling-related states may a thread be in?
Which scheduling decisions and events may trigger state changes?

Explain the following terms:

- Quantum,
- Priority boost,
- Priority Inversion.

Assignment 3.2:

Familiarize yourself with the following terms and concepts:

- Constructs for thread creation in the Win32-API
- Concepts for thread synchronization (Mutex, Semaphore, `WaitForMultipleObjects()`)

Assignment 3.3:

Modify your implementation of the `min_shell` command interpreter from assignment 2.4 in such a way, that it supports asynchronous execution of programs (i.e.; `min_shell` is not waiting for a process to complete before executing the next command). Asynchronous execution should be indicated by an & (ampersand) at the end of a line.

Extend `min_shell` with the following set of (`builtin`) instructions:

- `cd` – change the current working directory,
- `exit` – terminate the `min_shell` command interpreter,
- `wait` – wait for one or all asynchronously started processes.

Assignment 3.4:

Familiarize yourself with the concepts of memory management in the Windows 2000 OS. Answer the following set of questions:

- What is the *working set* of a process?
- What is a *page fault*? What is the distinction between *soft* and *hard page faults*?
- Describe the life cycle of a (virtual) memory page in Windows 2000. Explain the page list dynamics.